01-975

PATENT

10085929.022802

# IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

## APPLICATION FOR PATENT

## ON

## *OPTIMIZED READ PERFORMANCE METHOD USING METADATA TO PROTECT AGAINST DRIVE ANOMALY ERRORS IN A STORAGE ARRAY*

BY

KEITH W. HOLT
1522 KRUG CIRCLE
WICHITA, KANSAS 67230
CITIZEN OF USA

CERTIFICATE OF MAILING BY "EXPRESS MAIL"

"Express Mail" Mailing Label Number EV 013 244 704 US

Date of Deposit: February 28, 2002

I hereby certify that this correspondence is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 C.F.R. § 1.10 on the date indicated above and is addressed to Box Patent Application, Assistant Commissioner for Patents, Washington, D.C. 20231

BY: ________________
ReNea D. Berggren

# *OPTIMIZED READ PERFORMANCE METHOD USING METADATA TO PROTECT AGAINST DRIVE ANOMALY ERRORS IN A STORAGE ARRAY*

## FIELD OF THE INVENTION

**[0001]** The present invention generally relates to the field of storage array controllers, and particularly to an apparatus and method for efficiently detecting and recovering drive anomaly errors.

## BACKGROUND OF THE INVENTION

**[0002]** Reliable storage and the utilization of high availability storage arrays employing disk drives as the storage media are becoming evermore popular as data is stored electronically. Data is stored to, and retrieved from, an array of disks on a behalf of one or more host computer systems by storage array controllers. A major requirement of storage systems is the transfer and retrieval of data without error. Thus, storage systems and storage array controllers employ error detection and recovery algorithms to ensure data integrity.

**[0003]** A problem associated with high availability storage arrays is the return of incorrect data by a disk drive without an error indication. These types of errors may occur when writing data to, and reading data from, storage media. Drive anomaly protection is characterized by the assurance of data integrity, persistency and location. Data integrity assurance means that all bytes in a data block are stored, retrieved and transmitted correctly. Location assurance means that the data was stored to or retrieved from the correct physical location. Persistency assurance refers to whether data is actually written to media. Thus, detection of drive write anomalies involves the cross-checking of the integrity, persistency and location of data.

**[0004]** An approach known to the art to detect drive anomalies begins by storing a write sequence tracking metadata. The sequence information is stored on separate disks as metadata during write operations. On a subsequent read operation, the metadata is read from both disks and verified for consistency. The sequence information may be used to determine which drive is in error when the sequence information on the data drive is different from the parity drive. If the data drive is in error, the data is extracted from the parity drive via normal reconstruction techniques. The write sequence tracking scheme is implemented with a cyclic redundancy check (CRC) or similar form of error detection and correction code to provide data integrity protection. This provides data integrity assurance at a byte level to protect against drive anomaly errors in which the majority of data in the sector or sectors is correct. The CRC information may be stored as metadata along with the write sequence tracking information.

[0005] With this approach, write operations are tracked at two levels of granularity. The first level is when the scope of a write operation is limited to an individual drive plus the associated parity drive. In this case, the level of granularity is a data block such as the cache block size used to manage the storage controller's data cache. Each data block within a data stripe has its own separate revision number. The revision numbers of all data blocks are stored on the associated parity drive.

[0006] A second level of granularity is provided when all data blocks within a stripe are written. Each storage controller maintains a monotonically increasing value that is used to track full stripe writes on a storage controller basis. Tracking full stripe writes separately allows the controller to avoid having to perform a read-modify-write function an all of the associated data block revision numbers. When a full striped write occurs, all data block revision numbers are initialized to a known value.

[0007] The approach known to the art for detecting drive anomalies employs data read operation integrity cross-checks that require reading some form of metadata from a second drive. This effectively doubles a drive input/output (I/O) workload that results in a severe performance degradation for random read I/O profiles. In typical disk drives, the estimated impact is 40% decrease in I/O per second performance if the workload averages one or more I/O operations per drive. Further, this performance impact is relatively constant across a wide range of I/O sizes, drive seek ranges, and drive I/O queue depths. The decrease in performance is due to the fact that drive I/O per second performance does not double when workload doubles. Since only half of the drive I/O operations return user data, performance decreases from the host's perspective. Consequently, an apparatus and method for performing drive anomaly detection while optimizing random read performance is necessary.

## SUMMARY OF THE INVENTION

**[0008]** Accordingly, the present invention is directed to an apparatus and method for protecting against drive anomaly errors while optimizing random read performance. In one embodiment of the invention, data block persistency is explicitly verified when a data block is written. Data block integrity and location checks may be performed by reading data from a single drive. Through such a process, reading of metadata from a second drive is not required, thus providing anomaly protection without decreasing performance by increasing the drive I/O workload.

**[0009]** A drive anomaly protection apparatus of the present invention may employ a combination of a CRC and a location tag interleaved as metadata along with user data on a single drive. The location tag of the present invention may provide an indication of the logical block address or address range expected to be associated with a data block. Data persistency may be verified as part of each write operation through a write validation that ensures that data has been written to media. Data integrity assurance may be provided through the use of CRC information generated for each sector and stored as metadata during write operations. During read operations, the data block and CRC information may be retrieved from the drive and cross-checked for consistency. This may ensure that the correct data is retrieved. The location tag may also be checked during read operations by comparing it to the expected value that protects against retrieving data from an incorrect physical location.

**[0010]** It is to be understood that both the forgoing general description and the following detailed description are exemplary and explanatory only and are not restrictive of the invention as claimed. The accompanying drawings, which are incorporated in and constitute a part of the specification, illustrate an embodiment of the invention and together with the general description, serve to explain the principles of the invention.

## BRIEF DESCRIPTION OF THE DRAWINGS

**[0011]** The numerous advantages of the present invention may be better understood by those skilled in the art by reference to the accompanying figures in which:

**[0012]** FIG. 1 is a flow diagram depicting an exemplary method of the present invention for providing drive anomaly protection;

**[0013]** FIG. 2 is a flow diagram depicting an exemplary process for performing a read operation and recovery algorithm in accordance with the present invention; and

**[0014]** FIG. 3 depicts an embodiment of a storage controller of the present invention.

## DETAILED DESCRIPTION OF THE INVENTION

**[0015]** Reference will now be made in detail to the presently preferred embodiments of the invention, examples of which are illustrated in the accompanying drawings.

**[0016]** Referring to FIG. 1, a flow diagram 100 depicting an exemplary method of the present invention for providing drive anomaly protection is shown. The process may begin upon the verification of data persistency when a data block is written 110. This may ensure that stale data is not returned on a read operation. The commencement of a data read operation 120 follows the data persistency verification.

**[0017]** In accordance with the present invention, data integrity and location checks may be performed by reading data from a single drive 130. This may ensure that data has been retrieved properly from the correct physical location. In one embodiment of the invention, a data integrity test may be accomplished through a parity error detection algorithm such as a cyclic redundancy check (CRC). A location check may include a comparison of a location tag with an expected value.

**[0018]** Process 100 is advantageous as this may provide drive anomaly protection while optimizing random read performance. Applications that have a read/write ratio greater than one will experience improved performance compared to prior approaches since the process 100 of the present invention impacts drive workloads only for write operations. In addition, this performance benefit may provide increased benefits as the read/write ratio increases.

**[0019]** In one embodiment of the invention, persistency may be verified by means of a write validation that may ensure that data has been written to the media. The write validation may be synchronous or asynchronous to the completion of the write operation as long as the validation is complete prior to a host read of the data block. Synchronous validation may be accomplished by use of a drive "write verify with byte check" command. Asynchronous validation may be accomplished as a background operation by reading the data block and metadata via a drive "read with forced unit access" command. The results of the "read with forced unit access" command may be compared with data cached in the controller. A second method of asynchronous validation may allow the storage controller to return completion status to the host computer system prior to write validation.

**[0020]** Referring now to FIG. 2, a flow diagram depicting an exemplary process for performing a read operation and recovery algorithm 200 in accordance with the present invention is shown. The process may begin upon the commencement of a data read operation 210. Data and metadata from a data drive may be read into a controller's data cache 220. Metadata may be interleaved with user data for performance reasons. The granularity used for interleaving is flexible with regard to the algorithm.

**[0021]** CRC information may be generated for the data read from the data drive 230. The CRC information generated for the data read may be compared with the CRC information stored as metadata 240. This may be employed to ensure data integrity. Along with a CRC verification, a comparison of a location tag may be performed against an expected value 250. The location tag may be interleaved as metadata and provide an address range for the block of data. A determination of a CRC information and location tag match is performed 260. If the CRC information and location tag match, then the data from the data drive is error free 270. Thus, the data has been retrieved properly from the correct physical location. If the CRC information and the location tag do not match, then the data may be reconstructed 280.

**[0022]** Data integrity assurance at a byte level may be provided through the use of CRC information generated for each sector and stored as metadata during write operations. On read operations, the data block and CRC may be retrieved from the drive and cross-checked for consistency. This check may protect against partial data block corruption resulting from a misdirected write or any other anomaly that changes the data pattern on media. The location tag may be checked on read operations by comparing it to the expected value as described in step 250. This check may protect against full data block corruption resulting from a misdirected write.

**[0023]** Data recovery 280 may be implemented through multiple techniques. For example, data recovery may be possible through Redundant Array of Inexpensive Disks (RAID) parity schemes. Other types of data recovery may also be implemented with the algorithm 300 of the present invention by those with ordinary skill in the art without departing from the scope and spirit of the present invention. If a data drive is in error, data may be recovered using normal data reconstruction techniques. If a parity drive is in error, parity may also be rebuilt using parity repair techniques.

**[0024]** It should be understood that a various types of parity error information sets may be employed in accordance with the present invention to provide data integrity assurance without departing from the scope and spirit of the present invention. One type of parity error information set is CRC information. CRC information refers to an error detection method that uses parity bits generated by polynomial encoding of the data. It appends those parity bits to the data word. Receiving devices have decoding algorithms that detect errors in a data word. The decoding algorithm treats all bit streams as binary polynomials. CRC may be implemented through hardware, such as a shift register and exclusive OR gating circuitry. Software algorithms may also be employed to implement CRC.

**[0025]** The location tag of the present invention and referred to in the algorithm 200 may provide an indication of the logical block address or address range that is expected to be associated with the data block. As an example, the location tag may be set to the host logical block address associated with the start of the data block. The selection of a location tag for a given data block is flexible with regard to the algorithm.

**[0026]** In alternative embodiments of the present invention, CRC information may be generated and checked in multiple ways. In one embodiment of the invention, a hardware assist may be available for performance reasons, however, this is not required. When reading data from a disk, CRC information may be generated on the fly (simultaneous to receipt of data) or after the data has been received into the controller's data cache. A data integrity check may be performed with CRC information generated from well-known polynomials or alternative forms of error detection and correction codes. Other forms of error detection and correction code include, but are not limited to, Hamming codes, maximum-length codes, Bose-Chaudhuri-Hocquenghem Codes, Reed-Solomon Codes, and Convolutional Codes. Further, multiple ways of managing the CRC and location tag metadata may be available to those of ordinary skill in the art without departing from the scope and spirit of the present invention.

**[0027]** The read operation and data recovery algorithm 200 of the present invention is advantageous in many respects. The process of the present invention may allow symmetrical protection for data and parity drives. In one embodiment of the invention, there is not a distinction made between user data and parity information. CRC information and location tags may be generated, stored as metadata, and checked for the parity drive in a stripe just as it would be in a data drive. Another advantageous aspect of the read operation and data recovery algorithm 200 of the present invention is the flexibility in managing the metadata. The algorithm 200 may not have any explicit requirements on the granularity of the location tag or interleaving of metadata and user data. Further, there is not a requirement that either reads or writes of metadata and user data be atomic operations.

**[0028]** Referring now to FIG. 3, an embodiment of a storage controller 300 of the present invention is shown. In one embodiment of the invention, storage controller 300 may implement process 200 depicting an embodiment of a flow diagram for performing a read operation and recovery algorithm 200 of the present invention. Storage controller 300 may include boot read only memory (ROM) 310, random access memory (RAM) 320, processor 330, input/output interface 340, and a cache buffer 350. Input/output interface may receive/deliver data according to a desired protocol. Processor 330 may execute a program of instructions which may execute steps as shown in process 200 and may execute an algorithm on received data and transform the data into a desired protocol. It should be understood by those with ordinary skill in the art that process 200 may be implemented by other means than storage controller 300 without departing from the scope and spirit of the present invention.

**[0029]** It is believed that the system and method of the present invention and many of its attendant advantages will be understood by the forgoing description. It is also believed that it will be apparent that various changes may be made in the form, construction and arrangement of the components thereof without departing from the scope and spirit of the invention or without sacrificing all of its material advantages. The form herein before described being merely an explanatory embodiment thereof. It is the intention of the following claims to encompass and include such changes.